



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

mu

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/627,982

07/28/2003

Rajesh Bordawekar

YOR920030234US1

4112

48150

7590

10/05/2006

YOR.470

EXAMINER

INGBERG, TODD D

MCGINN INTELLECTUAL PROPERTY LAW GROUP, PLLC
8321 OLD COURTHOUSE ROAD
SUITE 200
VIENNA, VA 22182-3817

ART UNIT

PAPER NUMBER

2193

DATE MAILED: 10/05/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

10/627,982

Applicant(s)

BORDAWEKAR ET AL.

Examiner

Todd Ingberg

Art Unit

2193

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 28 October 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-69 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-69 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 2/6/04 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date <u>10/28/03</u> | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Claims 1 – 69 have been examined.

Drawings

1. Figures 7 and 8 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). Corrected drawings in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Information Disclosure Statement

2. The Information Disclosure Statement filed October 20, 2003 has been considered.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claims 67 and 68 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claims are directed to a signal directly or indirectly by claiming a medium and the Specification recites evidence where the computer readable medium is define as a "*wave*" (such as a carrier wave) . In that event, the claims are directed to a form of energy, which at present the office feels does not fall into a category of invention. The

following link on the World Wide Web is for the United States Patent And Trademark Office (USPTO) policy on 35 U.S.C. §101.

http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/guidelines101_20051026.pdf

Figure 8 is used to define the storage types. In the Specification “example” storage is the terms used. “Examples of” are not definitions.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1 – 3, 5, 7-13 and 17-69 are rejected under 35 U.S.C. 102(b) as being anticipated by the Template Software product line.

The **Template** Software product line contains:

The SNAP programming language

The Workflow Template

The Web Component

These three layered products work together.

The documentation sets for the products contains the following manuals.

SNAP released June 1997

SNAP Language Reference (Referred to as **REF** - Not used in this Office Action)

Using the SNAP Language (Referred to as **LANG** - Not used in this Office Action)

Using the SNAP Communication Component (Referred to as **COM**)

Art Unit: 2193

Using the SNAP Graphic User Interface Component (Referred to as **GUI** Not used in this Office Action)

Getting Started with SNAP (Referred to as **START** Not used in this Office Action)

Using the SNAP Display Editors (Referred to as **DISP** - Not used in this Office Action)

SNAP Class Library Reference (Referred to as **CLASS** - Not used in this Office Action)

Using the SNAP External Application Software Component (**EXT** - Not used in this Office Action)

Using the SNAP Development Environment (Referred to as **SNAP**)

SNAP Module Library Reference (Referred to as **MODU** Not used in this Office Action)

Using the SNAP Permanent Storage Component (Referred to as **PERM**)

Workflow released September 1997

Developing a WFT Workflow System (Referred to as **WFT** Not used in this Office Action)

Using the WFT Development Environment (Referred to as **ENV** Not used in this Office Action)

WFT Library Reference (Referred to as **WFTLIB** - Not used in this Office Action)

Web Component

Using the Web Component (Referred to as **WEB**)

Training Guides

SNAP Application Developer's Training Course (Referred to as **TRAINS** — Module 8 provided)

Workflow Template Training Course (Referred to as **TRAINW** - Not used in this Office Action)

Since, these products work together they constitute a single reference and can be used as the basis for a rejection based on anticipated by a product offering. Furthermore, with the 1997 press release announcing version 8.0 these considered prior art under *In re Epstein* 31 USPQ2d 1817 (decided August 17, 1994) with a 1997 release date despite the 1998 copyright date.

Art Unit: 2193

Claim 1

Template anticipates a method for detecting an error in an interaction between a plurality of software systems (COM, page 4-21 and PERM, pages 4-10 – 4-11), comprising: providing information about at least one of at least first and second software systems (COM, page 4-18, figure 4-4), and a mapping between at least a portion of said at least first and second software systems (PERM, Chapter 3 and SNAP, Chapter 7, page 7-5, Schema Editor – for mapping); and examining said at least one of said first and (COM, page 4-21) second software systems and said mapping to determine an error in an interaction between said at least first and second software systems (COM, page 4-21, code check for errors and SNAP Functions, pages 3-44 to 3-50).

Claim 2

The method of claim 1, wherein said information provided about at least one software system of said at least first and second software systems includes a schema of said at least one software system. As per claim 1.

Claim 3

The method of claim 1, wherein said information provided about at least one software system of said at least first and second software systems includes integrity constraints of the at least one software system. As per claim 1.

Claim 5

The method of claim 1, wherein said information provided about at least one software system of said at least first and second software systems includes code of the at least one software system. (COM, page 4-18, Figure 4-4).

Claim 7

The method of claim 1, wherein said information provided about at least one software system of said at least first and second software systems includes information about a sub-component of said at least one software system. (SNAP, class structure pages 3-37 to 3-39 and SNAP, page 5-19, #4).

Claim 8

The method of claim 1, wherein said information provided about at least one software system of said at least first and second software systems includes information about less than an entirety of said at least one software system. Interpreted to be Inheritance a principle of Object technology, SNAP, page 3-37.

Claim 9

The method of claim 1, wherein one of said first and second software systems comprises a database. SNAP, Chapter 6, Database Mapping Editor.

Art Unit: 2193

Claim 10

The method of claim 9, wherein said information provided about said, at least one software system of said at least first and second software systems includes schema information of said database. SNAP, Chapter 7, Schema Editor and Chapter 6, Database Mapping Editor.

Claim 11

The method of claim 9, wherein said information provided about said at least one software system of said at least first and second software systems comprises information about values in said database. SNAP, Attribute Sub-window, page 5-16.

Claim 12

The method of claim 1, wherein one of said first and second software systems comprises an application. As per claim 1.

Claim 13

The method of claim 12, wherein said information provided about said one of said first and second software systems includes programming language types. SNAP, pages 3-44 to 3-50.

Claim 17

The method of claim 1, wherein said mapping is provided explicitly. SNAP, Chapter 7.

Claim 18

The method of claim 1, wherein said mapping is inferred from said information about said at least first and second software systems. (COM, PAGES 5-8, 7-3, 8-2 Filter Functions – Provide the Ability to alter a value based on what was received).

Claim 19

The method of claim 1, wherein said error comprises an integrity constraint violation. As per claim 1.

Claim 20

The method of claim 1, wherein said error comprises a potential error representing a warning that an error may occur. (COM, PAGES 5-8, 7-3, 8-2 Filter Functions – Ability to trap violations prior to database interaction).

Claim 21

The method of claim 1, wherein said error comprises a definite error representing one of that an error will occur and that an error has occurred. As per claim 20.

Interpretation

One of is an OR

Claim 22

The method of claim 1, wherein said error is found prior to said interaction between the at least first and second software systems. As per claim 1.

Art Unit: 2193

Claim 23

The method of claim 1, wherein said error is found during said run time of the at least first and second software systems. As per claim 1.

Claim 24

The method of claim 1, wherein said error is found prior to any of said interaction between the at least first and second software systems As per claim 20, and during said runtime of the at least first and second software systems As per claim 1.

Claim 25

The method of claim 1, further comprising: inserting a check into at least one of said at least first and second software systems. (COM, PAGES 5-8, 7-3, 8-2 Filter).

Claim 26

The method of claim 25, wherein said check is inserted at a location directed by a programmer. As per claim 25, SNAP is a programming language and Template is a programming environment.

Claim 27

The method of claim 25, wherein said at least first and second software systems are checked after an interaction there between. Ability to produce Error as per claim 1.

Claim 28

The method of claim 25, wherein said at least first and second software systems are checked before an interaction there between. SNAP is a strongly typed language like C++ or JAVA, SNAPE types page 3-59 to 3-61. Type Checking is inherent in strongly typed languages. Type checking occurs before interaction of programs.

Claim 29

The method of claim 25, wherein said at least first and second software systems are checked prior to an end of a transaction. Both Filter functions as per claim 20 and type checking occur prior to end of a transaction.

Claim 30

The method of claim 25, further comprising: performing static analysis of said at least first and second software systems to at least one of simplify, eliminate, and approximate said check. SNAP, pages 3-40 to 3-43, Attributes definition and default.

Claim 31

The method of claim 1, further comprising: performing static analysis of said at least one of at least first and second software systems. As per claim 28, Type Checking.

Claim 32

Art Unit: 2193

The method of claim 3, further comprising: representing said integrity constraints of said at least first and second software systems in a common constraint model. SNAP, page 3-44, Schema and Object graph.

Claim 33

The method of claim 32, further comprising: analyzing said integrity constraints in said common constraint model. SNAP, page 3-44, Schema and Object graph.

Claim 34

The method of claim 33, further comprising: based on said analyzing, if an inconsistency is detected, then outputting an error. As per claim 1- errors.

Claim 35

The method of claim 31, further comprising modifying said integrity constraints in said common constraint model. SNAP, page 3-44, Schema and Object graph.

Claim 36

The method of claim 31, further comprising: generating a check from said integrity constraints. As per claim 20 – filter function and SNAP, page 3-44, edit function

Claim 37

The method of claim 31, further comprising: providing a shadow database in one of said at least first and second software systems, said shadow database containing partial knowledge of the other of said at least first and second software systems and being used to perform a check. SNAP, page 6-11, object filter.

Claim 38

The method of claim 37, wherein said partial knowledge includes partial knowledge of data values in said other of said at least first and second software systems. SNAP, page 5-16, Attribute window.

Claim 39

The method of claim 37, wherein said partial knowledge includes partial knowledge of non-existence of data values in said other of said at least first and second software systems. SNAP, page 5-16.

Claim 40

The method of claim 1, further comprising: reporting said error. (COM, page 4-21 and PERM, pages 4-10 – 4-11).

Claim 41

The method of claim 40, wherein said reporting comprises notifying before running at least one of said at least first and second software systems. Type checking as per claim 28.

Art Unit: 2193

Claim 42

The method of claim 40, wherein said reporting comprises notifying while running at least one of said at least first and second software systems. As per claim 1 – error.

Claim 43

The method of claim 40, wherein said reporting allows said one of said at least first and second software systems to address said error. As per claim 1.

Claim 44

The method of claim 40, wherein said reporting suggests how said error may be addressed. SNAP, page 3-45, Show Reference.

Claim 69

The method of claim 1, wherein said information provided about at least one software system of said at least first and second software systems includes a specification of said at least one software system. SNAP, pages 3-44, functions.

Claim 45

Template anticipates a method of detecting an error in a database interaction (COM, page 4-21), comprising: examining database code for database constraints (COM, pages 6-1 to 6-8, focus on 6-8 Remote Procedures and SNAP, Chapter 3, use of Functions to define methods to be performed – pages 3-44 to 3-50 – Note the COM reference mentions how to use the SNAP environment for Remote functions); examining application code for application-level constraints (Procedures and SNAP, Chapter 3, use of Functions to define methods to be performed – pages 3-44 to 3-58 – Rules and Daemons may also be used locally); and analyzing a mapping between said database code and said application code (COM, page 4-21), to determine an error in a database interaction (COM, page 4-21).

Claim 46

The method of claim 45, further comprising: generating a check in said application code for enforcing said database and application-level constraints. As per claim 45.

Claim 47

The method of claim 45, further comprising: forming a shadow database in said application code representing a portion of said database. SNAP, page 6-11, Object filter.

Claim 48

The method of claim 45, wherein said error comprises an integrity constraint violation. (COM, page 4-21, code check for errors and SNAP Functions, pages 3-44 to 3-50).

Claim 49

The method of claim 45, further comprising: inputting said database constraints and said application-level constraints, and said mapping into a common constraint model. SNAP, page 3-44, Schema and Object graph.

Claim 50

The method of claim 49, further comprising: before a program including said application and interacting with said database, is run and after said inputting, performing a static analysis to identify locations of where an error may arise. Type checking as per claim 28.

Claim 51

The method of claim 45, further comprising: after identifying said error and prior to running a program including said application, raising a notification. Type checking as per claim 28.

Claim 52

The method of claim 45, wherein, at runtime of said program, when an error is detected as occurring, raising a notification. (COM, page 4-21, code check for errors and SNAP Functions, pages 3-44 to 3-50).

Claim 67

A signal-bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of claim 45. SNAP, page 2-2.

Claim 53

Template anticipates a method of detecting an integrity constraint violation in a database interaction (COM, page 4-21 and PERM, pages 4-10 – 4-11), comprising: examining a database schema (PERM, Chapter 3 and SNAP, Chapter 7, page 7-5, Schema Editor – for mapping); examining an application type (SNAP, page 7-4, class and schema relation); and analyzing a mapping between, said database schema (PERM, Chapter 3 and SNAP, Chapter 7, page 7-5, Schema Editor – for mapping) and said application type (SNAP, page 7-4, class and schema relation), to determine whether an integrity constraint violation will occur in said database interaction with said application (COM, PAGES 5-8, 7-3, 8-2 Filter Functions – Ability to trap violations prior to database interaction).

Claim 54

The method of claim 53, wherein said database schema provides each of the integrated constraints defined in the database, and wherein said application type includes application code including integrity constraints defined therein. As per claim 45.

Claim 55

The method of claim 54, further comprising: generating a check in said application code for enforcing said database and application-level constraints. As per claim 45.

Claim 56

The method of claim 53, further comprising: forming a shadow database in application code representing a portion of said database. As per claim 37.

Art Unit: 2193

Claim 57

The method of claim 53, further comprising: inputting said database schema, said application type, and said mapping into a common constraint model. As per claim 32.

Claim 58

The method of claim 57, further comprising: before a program including said application and interacting with said database, is run and after said inputting, performing a static analysis to identify locations of where said integrity constraint violation may arise. As per claim 28.

Claim 59

The method of claim 53, further comprising: after determining said integrity constraint violation (As per claim 32) and prior to running a program including said application, raising a notification. As per claim 51.

Claim 60

The method of claim 53, wherein, at runtime of said program, when an integrity constraint violation is determined to occur, raising a notification. As per claim 52.

Claim 61

The method of claim 53, further comprising: analyzing a common constraint model receiving said database schema (as per claim 32), application type, and mapping, to determine an inconsistency between said database schema and said application type. As per claim 1.

Claim 62

The method of claim 61, wherein if no said inconsistency is determined, then taking all of the common constraints and analyzing the application code with respect to the common constraints for an error in the application code. As per claim 28.

Claim 63

The method of claim 62, wherein if no said error is determined in the application code, then inserting a check into said application code to enforce the constraints at runtime. As per claim 25.

Claim 68

A signal-bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of claim 53. SNAP, page 2-2.

Claim 64

Template anticipates a system of detecting an error in a database interaction (COM, page 4-21 and PERM, pages 4-10 – 4-11), comprising: a module for examining a database code for database constraints (COM, page 4-18, figure 4-4); a module for examining an application code for application-level constraints (COM, pages 6-1 to 6-8, focus on 6-8 Remote Procedures and SNAP, Chapter 3, use of Functions to define methods to be performed – pages 3-44 to 3-50 – Note the COM reference mentions how to use the SNAP environment for Remote functions)

Art Unit: 2193

; and an analyzing unit for analyzing a mapping between said database code and said application code (PERM, Chapter 3 and SNAP, Chapter 7, page 7-5, Schema Editor – for mapping), to determine an error in a database interaction (COM, page 4-21, code check for errors).

Claim 65

A system for detecting an integrity constraint violation in a database interaction (COM, pages 6-1 to 6-8, focus on 6-8 Remote Procedures and SNAP, Chapter 3, use of Functions to define methods to be performed – pages 3-44 to 3-50 – Note the COM reference mentions how to use the SNAP environment for Remote functions), comprising: a common constraint model for analyzing database schema (SNAP, page 3-44, Schema and Object graph) , application type , and a mapping between said database schema and said application type (PERM, Chapter 3 and SNAP, Chapter 7, page 7-5, Schema Editor – for mapping); and a determining unit for determining whether an integrity constraint violation will occur in said database interaction with said application (COM, page 4-21, code check for errors).

Claim 66

A method of constructing a program, comprising: detecting, in an application, portions of said application code that will or may raise a database integrity constraint violation during an application-database interaction during runtime (COM, page 4-21) , , the detecting including examining database schema (SNAP, Chapter 7, page 7-5, Schema Editor – for mapping), examining application type (SNAP, Chapter 7, page 7-5, Schema Editor – for mapping), and a mapping between the database schema (PERM, Chapter 3 and SNAP, Chapter 7, page 7-5, Schema Editor – for mapping) and the application type (PERM, Chapter 3 and SNAP, Chapter 7, page 7-5, Schema Editor – for mapping) ; inserting an integrity check notifying a programmer of such a definite or potential violation (Type checking as per claim 28) ; and completing the program (COM, PAGES 5-8, 7-3, 8-2 Filter Functions – Ability to trap violations prior to database interaction).

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 4, 6, 14-16 are rejected under 35 U.S.C. 103(a) as being unpatentable over

Template Software in view of UML and XML Schema, January 2002. (referred to as UX.

Motivation to Combine

Template teaches object modeling in non UML object modeler (SNAP, Chapter 3) and a hyper text language in HTML (Web, manual) which is not XML. UX teaches the use of the defacto standard UML and the more current XML language. Therefore, it would have been obvious to one of ordinary skill in the art at the time of invention because the use of UML and XML are more marketable.

Claim 4

The method of claim 1, wherein said information provided about at least one software system of said at least first and second software systems includes a unified modeling language (UML) model of said at least one software system. (UX, Abstract).

Claim 6

The method of claim 1, wherein said information provided about at least one software system of said at least first and second software systems includes at least one of: a schema of said at least one software system; integrity constraints of said at least one software system; a specification of said at least one software system; a unified modeling language (UML) model of said at least one software system; and code of said at least one software system. (UX, Abstract).

Claim 14

The method of claim 1, wherein one of said first and second software systems comprises one of an extensible markup language (XML) repository and an XML database..

Claim 15

The method of claim 14, wherein said information provided about said one of said first and second software systems includes one of XML schema information and XML document type definition (DTD) information. (UX, Abstract and Introduction).

Claim 16

The method of claim 2, wherein said schema includes XML schema information. (UX, Abstract).

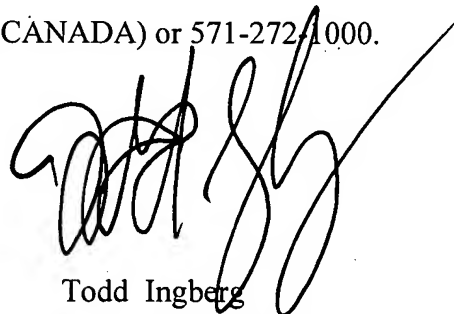
Correspondence Information

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Todd Ingberg whose telephone number is (571) 272-3723. The examiner can normally be reached on during the work week..

Art Unit: 2193

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (571) 272-3719. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

A handwritten signature in black ink, appearing to read 'Todd Ingberg', is written over the printed name and title.

Todd Ingberg
Primary Examiner
Art Unit 2193

TI